

Regularized Implicit Surface Reconstruction from Points and Normals

B. Mederos¹, M. Lage², S. Arouca², F. Petronetto², L. Velho¹, T. Lewiner² and H. Lopes²

¹Laboratório Visgraf
Instituto de Matemática Pura e Aplicada
{boris, lvelho}@impa.br

²Laboratório Matmídia – Departamento de Matemática
Pontifícia Universidade Católica do Rio de Janeiro
{sueni, mlage, fbpetro, tomlew, lopes}@mat.puc-rio.br

Abstract

We consider the problem of surface reconstruction of a geometric object from a finite set of sample points with normals. Our contribution is to present a new scheme for implicit surface reconstruction. Similarly to the multilevel partition of unity (MPU) method we hierarchically divide the domain obtaining local approximation for the object on each part, and then patch all together obtaining a global description of the object. Our new scheme uses ridge regression and weighted gradient one fitting techniques to get better stability on local approximations. The method behaves reasonably on sparse set of points and data with holes as those which comes from 3D scanning of real objects.

Keywords: Computer graphics, Implicit modeling, Surface reconstruction, Partition of unity, Ridge regression.

1. INTRODUCTION

Surface reconstruction is playing an important role in Computer Graphics [6]. Reconstruction is a very complex problem not only because the adjacency and proximity relations of data are unknown, but also because there are a lot adversities that need to be faced. The data in which these algorithms are applied come usually from 3D scanner. Currently this devices are able to handle real objects with increasing complexity and the resulting point clouds of the data acquisition contain fine details, rapid geometric variations, complex topology and sharp fea-

tures. However, the process of capturing the point cloud introduces sparse samples, holes (due to the occlusion of certain part of the object by other ones) and noise.

Several techniques have been studied to solve the surface reconstruction problem along the last two decades producing a variety of algorithms. Some methods are based on Delaunay triangulation concepts, among them, we can cite the works [13, 14, 15] for clean data and the works [12, 21] for noisy data. Other solutions are based on local parameterizations of shape like the Moving Least Square (MLS) approach [20, 10, 11]. Finally, there is also an important class of methods based on implicit function approximations. On this class, there are ones that use radial based functions (RBF) [8, 22, 25] and others based on domain decomposition schemes [17, 16, 19, 24, 3].

In this work we will focus on an implicit based scheme for surface reconstruction. Implicit Surface is a very useful representation for 3D objects, mainly because the inferred shape is computed by a formula which allows the computation of basic modeling operations in a relatively easy way [23]. Most of the boundary of man made objects are compose of several patches which can be approximated by algebraic surfaces. When the object's shape is complex, a common procedure is to elevate the algebraic degree in order to obtain more precision on the approximation. However, in this case due to the ill posedness of this fitting some spurious connected components appear on the reconstructed surface [5]. An alternative solution is to decompose the domain hierarchically in compact parts and obtain local approximation for the object in

each part, and then patch all together in order to obtain a global description of the object. A practical scheme that uses such solution is the method of multilevel partition of unity implicit (MPU) [17]. It provides an adaptive error-controlled approximation of the signed distance function from the surface.

Problem description. Given a finite set of points \mathcal{P} in \mathbb{R}^3 sampled from a surface \mathcal{S} in the space, also assume that for each point $p \in \mathcal{P}$ the unitary normal vector n to the surface at p is given. The objective of this work is to obtain an implicit function $F : \mathbb{R}^3 \rightarrow \mathbb{R}$ such that the isosurface of level zero, $F^{-1}(0)$, approximates adaptively \mathcal{S} using local error-control.

Contributions. The reconstruction scheme proposed in this paper is an extension to 3D surface reconstruction of the work [18] proposed to tackle the 2D case of curve reconstruction. In this extension we introduce a weighted local algebraic approximation which produces a substantial improvement compared to the original two dimensional method. Our algorithm is based on the main ideas of MPU [17] that hierarchically subdivides the domain in several parts and later computes local shape approximations on each part. Our method presents a different strategy for the local shape approximations and improves its numerical stability by the use of the *ridge regression* technique and *weighted gradient one fitting*. As a consequence, it avoids the generation of spurious connected components on the reconstructed implicit surface.

Paper outline. Section 2 introduces some basic concepts. Section 3 describes the main ideas of the original MPU method. Section 4 describes some schemes for implicit surface fitting. Section 5 introduces our new method and the implementation details. Section 6 shows some results. Finally, Section 7 concludes and proposes future directions of this work.

2. IMPLICIT SURFACE AND PARTITION OF UNITY

2.1. IMPLICIT SURFACE

A subset $\mathcal{O} \subset \mathbb{R}^3$ is called an *Implicit Surface* if there is a function $F : U \rightarrow \mathbb{R}$, $\mathcal{O} \subset U$, and a real number $c \in \mathbb{R}$ such that $\mathcal{O} = F^{-1}(c)$. The implicit surface $F^{-1}(c)$ is *regular* if F is differentiable and satisfies the condition that at each point $\mathbf{x} \in F^{-1}(c)$ the gradient of F at x does not vanish.

A *polynomial of degree d* defined on \mathbb{R}^3 is a function $P_d : \mathbb{R}^3 \rightarrow \mathbb{R}$ given by the following expression:

$$P_d(x, y, z) = \sum_{0 \leq i+j+k \leq d} a_{i,j,k} x^i y^j z^k.$$

An *algebraic surface of degree d* is the implicit surface $P_d^{-1}(0)$. It is convenient to adopt a suitable notation for P_d . We adopted the vectorial one proposed by Tasdizen et al. in [18], which is:

$$P_d(x, y, z) = \mathbf{v}_{(x,y,z)}^t \mathbf{a}, \quad (1)$$

where

$$\mathbf{v}_{(x,y,z)} = [1 \cdots x^d y \cdots x^{d-1} y \cdots z \cdots z^d]^t \quad (2)$$

and

$$\mathbf{a} = [a_{0,0,0} \cdots a_{d,0,0} a_{0,1,0} \cdots a_{d-1,1,0} \cdots a_{0,0,1} \cdots a_{0,0,d}]^t$$

The elements of the vector $\mathbf{a} \in \mathbb{R}^l$ are the coefficients $a_{i,j,k}$ ($0 \leq i, 0 \leq j, 0 \leq k$, and $0 \leq i + j + k \leq d$) of P_d , and the elements of the vector $\mathbf{v}_{(x,y,z)} \in \mathbb{R}^l$ are the monomials of P_d . The dimension l of the vectors \mathbf{a} and $\mathbf{v}_{(x,y,z)}$ depends on the degree d and is obtained by the following expression: $l = \frac{(d+1)(d+2)(d+3)}{6}$.

2.2. PARTITION OF UNITY

A *partition of unity* (PU) [1],[4] is a mathematical tool very useful to combine local approximation in order to define a global one. Important properties such as the global maximal error and the convergence order could be inherited from the local approximation. The basic ideas behind the global approximation construction using partition of unity are the following:

- i. divide the domain in parts,
- ii. obtain a local approximation for each part using a subset of the data that belongs to it,
- iii. obtain a global approximation by the use of a weighted combination of local solution through the use of smooth non-negative functions that correspond to the weights. In each point of the domain, the sum of these weight functions should be one.

More precisely, consider a compact domain $\Omega \subset \mathbb{R}^3$ and denote by $\{\varphi_i\}_{i=1,\dots,n}$ the set of non-negative functions with compact support such that:

$$\sum_{i=1}^n \varphi_i(x, y, z) \equiv 1 \text{ for all points } (x, y, z) \in \Omega.$$

Let \mathcal{F}_i be a set of functions defined in $\text{supp}(\varphi_i)$. Each function in \mathcal{F}_i represents a local approximation for the points of \mathcal{P} that belong to $\text{supp}(\varphi_i)$. A global approximation for the function $f : \Omega \rightarrow \mathbb{R}$ could be obtained as following:

$$f(x, y, z) \approx \sum_{i=1}^n \varphi_i(x, y, z) f_i(x, y, z). \quad (3)$$

where $f_i \in \mathcal{F}_i$. Consider $\{w_j\}_{j=1,\dots,n}$ a set of non-negative functions with compact support such that:

$$\Omega \subset \bigcup_{j=1}^n \text{supp}(w_j).$$

The partition of unity functions φ_i could be generated by the following equation:

$$\varphi_i(x, y, z) = \frac{w_i(x, y, z)}{\sum_{j=1}^n w_j(x, y, z)} \quad (4)$$

The main idea of partition of unity could be resumed by the equations (3) and (4). Such equations form the basis of the algorithm Multilevel Partition of Unity Implicit (MPU) proposed by Ohtake et al. in [17].

3. MULTILEVEL PARTITION OF UNITY

The method called *Multilevel Partition of Unity* (MPU) was proposed by Ohtake et al. in [17] originally to build an implicit surface approximation of a set of points and normals in \mathbb{R}^3 . The MPU uses a partition of unity to obtain a global implicit surface approximation for the boundary of the object combining local approximations. It uses an *octree* as an hierarchical scheme to guide the domain subdivision.

Follows a concise description of how the MPU builds an implicit function that globally approximates the points.

The method initially centers the point of \mathcal{P} at the origin. After that, the points are scaled in such a way that the square $\Xi = [-1, 1]^3$ contains all points of \mathcal{P} . We will adopt the same name \mathcal{P} for the set of points after these two transformations.

The method builds an *octree* by the use of recursive procedure where the subdivision of each node is controlled by the error of the local approximation. In other words, the refinement criteria for a node i of the *octree* consists of computing the local error of the approximation and when this error is greater than a given tolerance, then the node is subdivided in eight new nodes and recursively the same test is again used for each one of its child nodes.

Each node i on the *octree* is associated to a weight function w_i with a compact support that is used for the partition of unity global approximation. The compact support of w_i is defined as a circle of radius r_i centered on the middle of the node i . Such radius is chosen proportionally to the size of the diagonal of the square corresponding to the node i , denoted by d_i .

In the MPU original method, a quadratic spline function $b : \mathbb{R} \rightarrow \mathbb{R}$ is used to build the weight function w_i :

$$w_i(x, y, z) = b\left(\frac{3 \left| (x, y, z) - c_i \right|}{2r_i}\right) \quad (5)$$

where c_i is the center of the corresponding node i in the *octree*. The value of w_i is zero outside the support region. According to the equation (4), the partition of unity function φ_i associated to the node i is defined as:

$$\varphi_i(x, y, z) = \frac{w_i(x, y, z)}{\sum_{j=1}^{n_i} w_j(x, y, z)}, \quad i = 1 \dots n_l \quad (6)$$

where n_l is the number of leaves on the *octree*.

The MPU method uses a quadric (degree 2 polynomial function) to locally approximate the signed distance function to the boundary of the object. According to the distributions of the normal in a node of the octree either a 3D quadric or bivariate quadric polynomial are used to approximate the local shape. To find the coefficients of the quadric function $Q_i : \text{supp}(\varphi_i) \subset \mathbb{R}^3 \rightarrow \mathbb{R}$ for the node i , Ohtake et al. [17] used a least squares scheme that will be reviewed in subsection 4.1.

At each node, a least squares problem is solved considering only the points on \mathcal{P} that belong to its support region as input. Sometimes (especially when the density of \mathcal{P} is not uniform) the circle of radius r_i of a node i doesn't contain a sufficient number of points to estimate robustly the quadric that approximates such points. If the number of points on the support region of a node is not sufficient to solve the minimization problem, then they adopted a solution that increases the radius of the support region until such minimal condition is guaranteed.

They suppose that the surface \mathcal{S} from where the points are sampled is a level 0 isosurface of a function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$. Then, they can use the partition of unity equation (7) to obtain a $F : \mathbb{R}^3 \rightarrow \mathbb{R}$ that globally approximates f :

$$f(x, y, z) \approx F(x, y, z) = \sum_{i=1}^{n_l} \varphi_i(x, y, z) Q_i(x, y, z). \quad (7)$$

For more information of this method, see [17].

4. IMPLICIT SURFACE FITTING

Follows the description of some methods to obtain an implicit surface approximation.

4.1. MINIMIZING THE ALGEBRAIC DISTANCE

Suppose that a set of q points $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_q\} \subset \mathbb{R}^3$ sampled from a 3D surface \mathcal{S} is given. A simple way to obtain an algebraic surface $P_d^{-1}(0)$ that approximates \mathcal{S} is to minimize the total sum of the squared algebraic distance, denoted by e_{alg} , from each point $\mathbf{p}_i \in \mathcal{P}$ to the curve $P_d^{-1}(0)$. Using the vectorial representation of P_d in (1), we can write e_{alg} as:

$$e_{alg} = \mathbf{a}^t \left(\sum_{i=1}^q \mathbf{v}_i \mathbf{v}_i^t \right) \mathbf{a}, \quad (8)$$

where \mathbf{v}_i corresponds to the evaluation of the vector \mathbf{v} , see equation (2), at the points $p_i = (x_i, y_i, z_i)$. This minimization is subject to the constraint that the sum of the squared coefficients of P_d is equal to one ($\|\mathbf{a}\|^2 = 1$). Note that without this constraints the minimum of e_{alg} is reached at the null vector. A more robust approximation measure is to consider the weighted algebraic distance

$$e_{alg} = \frac{\mathbf{a}^t (\sum_{i=1}^q w_i \mathbf{v}_i \mathbf{v}_i^t) \mathbf{a}}{\sum_{i=1}^q w_i},$$

where $w_i = w(p_i) \geq 0$ is a set of weight and $w(x, y, z)$ is a positive weight function such that $w(x)$ approaches to zero when $\|x\|$ goes to infinity or a compactly support function. To improve the notation, define the matrices \mathbf{M} of size $l \times q$ and \mathbf{S} of size $l \times l$ as follows:

$$\mathbf{M} = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_q],$$

and

$$\mathbf{S} = \mathbf{M} \mathbf{M}^t = \frac{\sum_{i=1}^q w_i \mathbf{v}_i \mathbf{v}_i^t}{\sum_{i=1}^q w_i}.$$

Thus, the minimization problem we have to solve is the following:

$$\begin{aligned} \min_{\mathbf{a}} \{ \mathbf{a}^t \mathbf{S} \mathbf{a} \}, \\ \text{subject to: } \|\mathbf{a}\|^2 = 1. \end{aligned}$$

By the use of the Lagrange multiplier λ , the constrained minimization problem is rewritten as:

$$\min_{\mathbf{a}} \{ \mathbf{a}^t \mathbf{S} \mathbf{a} + \lambda (\mathbf{a}^t \mathbf{a} - 1) \}.$$

Its solution is the unitary eigenvector of S associated to the eigenvalue of S with smallest value [7].

Although this method is invariant to affine transformations [7], it has some delicate problems. Its results are sensitive to small perturbations on the input data. Moreover, the algebraic surface $P_d^{-1}(0)$ doesn't consider the points continuity. Thus, it could generate undesired connected components or glue components that are originally separated. For more details see [2, 18, 7].

4.2. GRADIENT ONE FITTING

To avoid the problems of continuity and of sensitivity caused by small perturbation on the input data [2] proposed a new method that considers not only the set \mathcal{P} but also a given set of normal unitary vectors $\mathcal{N} = \{\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_q\}$, where the vector \mathbf{n}_i is the normal vector to the surface \mathcal{S} at the sampled point $p_i \in \mathcal{P}$.

The gradient

$$\nabla P_d(p_i) = \left(\frac{\partial P_d}{\partial x}(p_i), \frac{\partial P_d}{\partial y}(p_i), \frac{\partial P_d}{\partial z}(p_i) \right)$$

of P_d at the point $p_i = (x_i, y_i, z_i)$, will be used by this method to approximate the direction of the given normal. It is important to notice that if the gradient of P_d at $p_i =$

(x_i, y_i, z_i) is not equal to zero, then it is perpendicular to the tangent plane of the level surface P_d that pass through the point p_i .

The *gradient one fitting* (GOF) method [2] is in fact an unconstrained least square problem. We have introduced a modification in the original GOF method that consist of a weighted least square problem as follows

$$e_{grad} = \sum_{i=1}^q w_i [(P_d(x_i, y_i, z_i))^2 + \mu (\mathbf{n}_i^t \nabla P_d(x_i, y_i, z_i) - 1)^2] \quad (9)$$

where μ is the weight given to the terms that approximates the normals and w_i is a set of weight depending of each p_i . In order to follow the vectorial representation of P_d , we define the following matrices and vectors:

- The matrix $\mathbf{D}_i = \begin{bmatrix} \frac{\partial \mathbf{v}_i}{\partial x} & \frac{\partial \mathbf{v}_i}{\partial y} & \frac{\partial \mathbf{v}_i}{\partial z} \end{bmatrix}$ of size $l \times 3$:

- The gradient vector ∇P_d :

$$\nabla P_d = \nabla(\mathbf{v}_i^t \mathbf{a}) = (\mathbf{D}_i)^t \mathbf{a}.$$

- The matrix \mathbf{S}_N of size $l \times l$:

$$\mathbf{S}_N = \sum_{i=1}^q w_i \mathbf{D}_i \mathbf{n}_i \mathbf{n}_i^t \mathbf{D}_i^t.$$

- The vector \mathbf{g}_N of size l :

$$\mathbf{g}_N = \sum_{i=1}^q w_i \mathbf{D}_i \mathbf{n}_i.$$

Therefore, the weighted GOF optimization problem can be rewritten as:

$$\min_{\mathbf{a}} \{ \mathbf{a}^t (\mathbf{S} + \mu \mathbf{S}_N) \mathbf{a} - 2\mu \mathbf{a}^t \mathbf{g}_N + \mu q \}.$$

And its solution is obtained by solving the following system of linear equations:

$$(\mathbf{S} + \mu \mathbf{S}_N) \mathbf{a} = \mu \mathbf{g}_N. \quad (10)$$

4.3. MPU 3D FITTING

The *MPU 3D fitting* is based on the minimization of the squared algebraic distance e_{alg} (8) plus an energy term e_{mpu} which in some way tries to favor a local approximation of the signed distance function to \mathcal{P} by 3D quadric.

The e_{mpu} depends on the set of auxiliar points $\mathcal{Q} = \{q_1, q_2, \dots, q_m\}$ which is a subset of the vertices of the node in the subdivision octree, and this implies that the maximum cardinality of \mathcal{Q} is eight. Following [17] a vertex q of the node is reliable for computing the approximated average signed distance, $d = 1/3 \sum_{i=1}^3 \mathbf{n}_i(q - p_i)$ (where p_1, p_2 and p_3 are its three near sample points on \mathcal{P}) if $\mathbf{n}_1(q - p_1)$, $\mathbf{n}_2(q - p_2)$ and $\mathbf{n}_3(q - p_3)$ have the same sign.

Then the energy e_{mpu} is defined as

$$e_{mpu} = \frac{1}{m} \sum_{q_j \in \mathcal{Q}} (P_d(q_j) - d_j)^2, \quad (11)$$

where d_j is the approximated average signed distance of q_j to \mathcal{P} . The combination of e_{mpu} with GOF is a good way to determine local shape approximation which takes advantage of the benefits of (9) and (11)

$$e = e_{grad} + e_{mpu}. \quad (12)$$

The minimization of (12) conduces to the following linear optimization problem

$$\min_{\mathbf{a}} \{ \mathbf{a}^t (\mathbf{S} + \mu \mathbf{S}_N + \mu_1 \mathbf{S}_{MPU}) \mathbf{a} - 2 \mathbf{a}^t (\mu \mathbf{g}_N + \mu_1 \mathbf{g}_{MPU}) \}, \quad (13)$$

where \mathbf{S}_{MPU} and \mathbf{g}_{MPU} are defined as follows:

$$\mathbf{S}_{MPU} = \sum_{q_j \in \mathcal{Q}} \mathbf{v}_{q_j} \mathbf{v}_{q_j}^t \quad (14)$$

and

$$\mathbf{g}_{MPU} = \sum_{q_j \in \mathcal{Q}} d_j \mathbf{v}_{q_j}. \quad (15)$$

The minimization problem (13) conduces to the following linear system

$$(\mathbf{S} + \mu \mathbf{S}_N + \mu_1 \mathbf{S}_{MPU}) \mathbf{a} = \mu \mathbf{g}_N + \mu_1 \mathbf{g}_{MPU}. \quad (16)$$

4.4. RIDGE REGRESSION

When the matrix $\bar{\mathbf{S}} = \mathbf{S} + \mu \mathbf{S}_N + \mu_1 \mathbf{S}_{MPU}$ doesn't have a maximal rank or is ill conditioned then the technique called *ridge regression* (RR) can be used. Statisticians use it frequently to remove the collinearity of the input data. The first proposal to obtain algebraic surfaces that fits better was done by Tasdizen et al. in [18]. The RR technique basically modifies the optimization problem of the *gradient one fitting* method by adding a new term:

$$\min_{\mathbf{a}} \{ \mathbf{a}^t \bar{\mathbf{S}} \mathbf{a} - 2 \mu \mathbf{a}^t \mathbf{g}_N - 2 \mu_1 \mathbf{a}^t \mathbf{g}_{MPU} + \mu q + \kappa \mathbf{a}^t \mathbf{\Delta} \mathbf{a} \},$$

where $\mathbf{\Delta}$ is a diagonal matrix of size $l \times l$ and the real constant κ determines the weight given to the new term. The minimization problem solution is obtained by solving the following system of linear equations:

$$(\bar{\mathbf{S}} + \kappa \mathbf{\Delta}) \mathbf{a} = \mu \mathbf{g}_N + \mu_1 \mathbf{g}_{MPU}$$

5. PROPOSED RECONSTRUCTION SCHEME

We propose in this section a new method for implicit surface fitting. It combines the MPU scheme and the *gradient fitting one* and *ridge regression* methods to improve the implicit surface approximation.

Data input. Our algorithm considers as input data a set of q points \mathcal{P} , and a set of the corresponding q unitary normal vectors \mathcal{N} .

We are assuming that the points of \mathcal{P} have been translated in such a way that the center of mass is the origin of the coordinate system, and also that they have been scaled in such a way that all points are contained in the square $\Xi = [-1, 1]^3$. Such square Ξ is the starting region for the hierarchical adaptive space subdivision guided by the use of an *octree* data structure.

Support regions. We use the same *octree* support region scheme of the MPU method. The support region for each node i is the disk of radius $r_i = \alpha d_i$ centered at c_i .

Local approximations. In our method, we adopt the *mpu 3d fitting*, *gradient one fitting* and *ridge regression* techniques to obtain the coefficients of a degree d algebraic function P_{d_i} for the local approximation at the node i . The local approximation is only computed when the node contains sample points on its support region. To do this task, we consider the set of points $\mathcal{P}_i = \{p_{j_1}, \dots, p_{j_{a_i}}\} \subset \mathcal{P}$ that are on the support region of the node i , in case that we have enough points (a number bigger than $N_{min} = 15$), otherwise we grow the support region until we get a minimum number of points. To run the RR method, we have also to equip each point of \mathcal{P}_i with its unitary normal vectors. Thus, using the same indexes of these vectors we construct the sets $\mathcal{N}_i \subset \mathcal{N}$ for each node i .

To determine the local approximation to the shape of the surface on the node i we take a similar strategy to the MPU algorithm, i.e, two types of local approximations are computed: 1) a 3D quadric, 2) a bivariate quadratic polynomial in local coordinates. The first is used to approximate parts of the surface composed of more than one surface sheet and the second to approximate a local smooth patch. To determine which of these kinds of approximation we will compute we apply the same method proposed in [17] which roughly speaking is based on computing an average normal direction on \mathcal{N}_i , if the maximum deviation of the normals to the average normal n_{ave} is bigger than $\pi/2$ then we compute 1) otherwise we compute 2), for more detail, see [17].

In the first case the coefficient of 3D quadric are determined minimizing the objective function:

$$\mathbf{a}^t (\mathbf{S}_i + \mu \mathbf{S}_{N,i} + \mu_1 \mathbf{S}_{MPU,i} + \kappa I) \mathbf{a} - 2 \mathbf{a}^t (\mu \mathbf{g}_{N,i} + \mu_1 \mathbf{g}_{MPU,i})$$

where the matrices \mathbf{S}_i , $\mathbf{S}_{N,i}$, $\mathbf{S}_{MPU,i}$ and the vectors $\mathbf{g}_{N,i}$, $\mathbf{g}_{MPU,i}$ are computed for the node i using the expressions presented in Section 4. However, we consider as input for this computation the sets \mathcal{P}_i and \mathcal{N}_i . The solution to the minimization problem is obtained by solving

the following system of linear equations:

$$(\mathbf{S}_i + \mu \mathbf{S}_{N,i} + \mu_1 \mathbf{S}_{MPU,i} + \kappa I) \mathbf{a} = \mu \mathbf{g}_{N,i} + \mu_1 \mathbf{g}_{MPU,i}$$

In the second case a bivariate quadratic polynomial is determined with the domain being the plane Π_i orthogonal to n_{ave} passing through the centroid of the points in \mathcal{P}_i minimizing

$$\sum_{p_j \in \mathcal{P}_i} w_j [(\tilde{P}_d(\tilde{x}_j, \tilde{y}_j) - \tilde{z}_j)^2 + \mu(n_j^t \tilde{n}_j - 1)^2] \quad (17)$$

where

$$\tilde{n}_j = (1, 0, \frac{\partial P_d(\tilde{x}_j, \tilde{y}_j)}{\partial x}) \times (0, 1, \frac{\partial P_d(\tilde{x}_j, \tilde{y}_j)}{\partial y})$$

and $(\tilde{x}_j, \tilde{y}_j, \tilde{z}_j)$ are the coordinates of the points $p_j \in \mathcal{P}_i$ in an orthogonal system of coordinates (u, v, n_{ave}) where u, v is an orthogonal base in the plane Π_i . $\tilde{P}_d(x, y)$ is a bivariate polynomial of degree d , the minimization of (17) conduces to solving a 2×2 linear system of the same type of (10).

Octree construction. The *octree* is built using a recursive procedure, whose refinement criteria is the local approximation error. Consider a tolerance ϵ for the local error. The condition that determines whether node i of the *octree* at level l_i should be refined is the following boolean expression:

$$(e_i \geq \epsilon) \text{ and } (l_i < l_{max}) \text{ and } (q_i > N_{min}),$$

where the local approximation error e_i at node i is the widely known Taubin error metric and also the mean squared algebraic distances from the points \mathcal{P}_i to the obtained surface has been considered. It seems that the Taubin error metric produces good result in less time.

Parameters of the method. In conclusion, the parameters of the method are the following:

- $d \in \mathbb{N}$: degree of the algebraic surface.
- $l_{max} \in \mathbb{N}$: maximum level for the *octree*
- $\alpha \in \mathbb{R}$: constant that multiplies the diagonal size of the node to obtain the radius of the support disk.
- $\mu \in \mathbb{R}$: weight given to the GOF term on the objective function.
- $\mu_1 \in \mathbb{R}$: weight given to the MPU term on the objective function.
- $\kappa \in \mathbb{R}$: weight given to the ridge regression additional term on the objective function.
- $\epsilon \in \mathbb{R}$: threshold value for the refinement condition controlled by the local approximation error.

Notice that with this set of parameters we can unify several methods presented in this paper. For example: if we want to run the original GOF method we have to assign $l_{max} = 0$ and $\kappa = 0$; if we want the original RR method we have only to assign $l_{max} = 0$.

Global approximation and function evaluation. We have supposed that the surface can be written as $S = f^{-1}(0)$ for some function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$. Thus, by the use of the partition of unity equation (7) we can obtain a function $F : \mathbb{R}^3 \rightarrow \mathbb{R}$ that globally approximates f :

$$f(x, y, z) \approx F(x, y, z) = \sum_{i=1}^{n_i} \varphi_i(x, y) P_{d_i}(x, y).$$

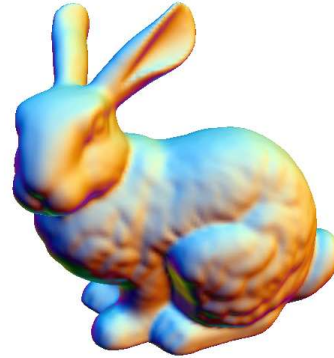


Figure 1. Bunny model, with parameters $l_{max} = 30$, $\mu = 0.01$, $\mu_1 = 1.0$, $\epsilon = 0.0005$ and $\kappa = 0.0$.

6. RESULTS

Our method has been tested with several sparse data sets, Figures 1, 3 and 4 show the reconstruction of the points sets sampled from the bunny, knot and armadillo surface respectively using our method. In all these examples the ridge regression parameter have been fixed to $\kappa = 0.0$. The surfaces were tiled using an implementation of Topological Marching Cubes [9]. The time involved is very similar to the original MPU [17]: within 1% for the optimization, and identical for the evaluation. The typical parameters of our method are $\mu_1 = 1.0$, $\mu = 0.01$, $\kappa = 0.001$, $l_{max} \in [20, 30]$, $\epsilon \in [0.0005, 0.005]$ and $\alpha = 0.75$.

In Figure 2 in the left we have the result of our implementation of the MPU algorithm on the dragon data set in which we can see a small connected component and some artifacts on the model surface, the right picture shows the results of our method (MPU + *gradient one fitting* + *ridge regression*) we gain more numerical stability and we are able to remove the spurious small connected component and some of these artifacts. We tested the algorithm on this data set for different values of κ ranges from 0.005,

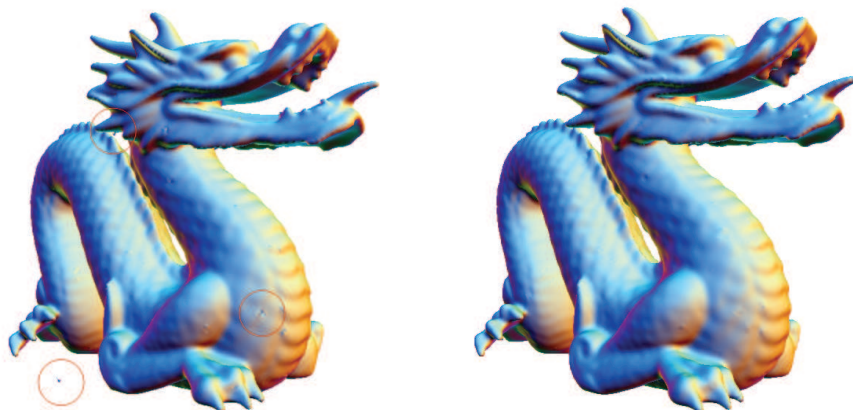


Figure 2. Dragon model: (left) result of the MPU algorithm which corresponds to our method for parameters $l_{max} = 30$, $\mu = 0.0$, $\mu_1 = 1.0$, $\epsilon = 0.0005$ and $\kappa = 0.0$. (right) result for parameters $l_{max} = 30$, $\mu = 0.01$, $\mu_1 = 1.0$, $\epsilon = 0.0005$ and $\kappa = 0.001$: small artifacts are removed

0.001, 0.0005, 0.0002 and for these values we were able to remove this small connected component and some of these artifacts, for $\kappa = 0.0001$ a small spurious component appear on the reconstructed surface. The use of ridge regression is necessary to remove the artifacts and the connected component a combination of (MPU + *gradient one fitting*) is not enough to get a good result.

The effect of gradient one fitting is illustrated on the squirrel model (Figure 5), with incomplete point cloud, similarly to the usual output of laser scanner: the whole bottom part and details of the eyes and on the top are missing. Subfigures c and e show the results of the only MPU method. Observe that, due to the big hole on the head, the MPU alone generated a bump and also inaccuracy on the right eye. The combination of the MPU with gradient one fitting (Subfigures d and f) is able to reconstruct the model without bump on the head and has a greater accuracy on the eyes. In this example the use of the ridge regression does not significantly alter the results with gradient one fitting. In general, the use of the gradient one fitting can be useful in the presence of small holes since it uses the neighboring normals.

7. CONCLUSIONS AND FUTURE WORKS

We proposed a new method that combines two powerful techniques: the *weighted gradient one fitting + ridge regression* and the *multilevel partition of unity*. On one side, the *ridge regression* method has been considered by the pattern analysis community as one that gives a better fitting, since it tries to have a correct topology on the surface reconstruction. However, when the surface has a complex shape it is necessary to elevate the degree of the algebraic surface to get a good result. On the other side, the *multilevel partition of unity* is an implicit method

that is now one of the most important reconstruction techniques. In order to compute local approximations, it uses a complicate objective function. Thus, our surface reconstruction scheme not only takes the advantage of these two well recognized methods, but also unifies those methods in a simple setting.

We plan to continue this work in several directions: one possible direction is to determine an approximated tangent plane T to the samples on the node of the tree and over this plane we consider the surface as a height field and determine a bivariate local approximation $Q(x, y) : T \rightarrow \mathbb{R}$. Also over T we can consider better approximations like for example using wavelets method in order to be able to faithfully reproduce the oscillations and details (texture) on each region.

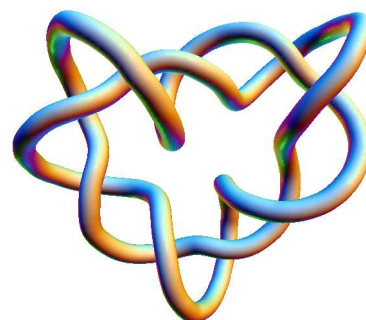


Figure 3. Knot model, with parameters $l_{max} = 30$, $\mu = 0.01$, $\mu_1 = 1.0$, $\epsilon = 0.0005$ and $\kappa = 0.0$.

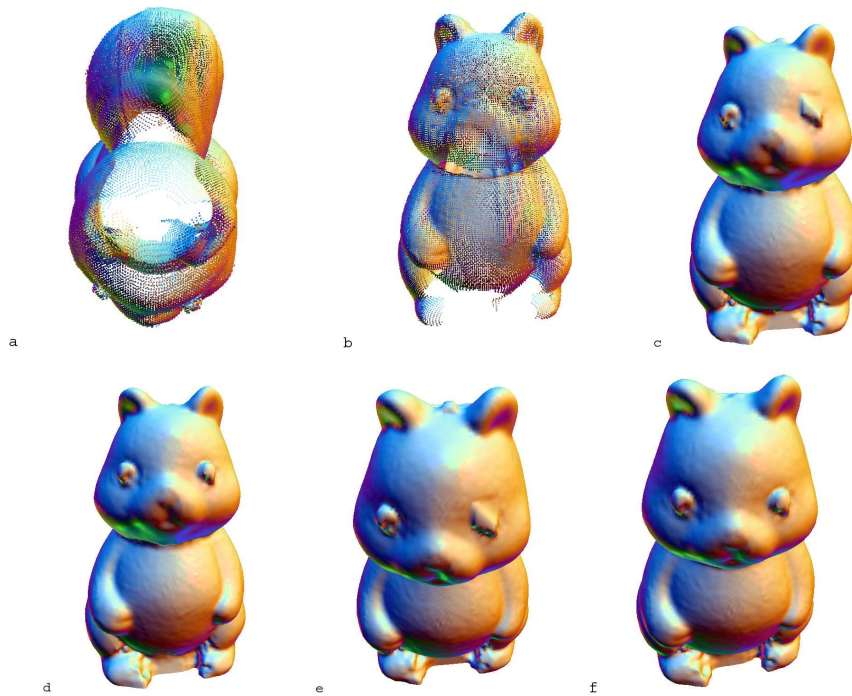


Figure 5. Squirrel model. Figures a) and b) show the Squirrel point cloud, notice the holes on the top and the front, also there are small holes in the eyes. Pictures c) show the results of the MPU method $l_{max} = 20$, $\mu = 0.0$, $\mu_1 = 1.0$ and $\epsilon = 0.005$ and picture d) shows the combination of the MPU and gradient one fitting with parameters ($l_{max} = 20$, $\mu = 0.01$, $\mu_1 = 1.0$ and $\epsilon = 0.005$) respectively. Pictures e) and f) show the results of pictures c) and d) from some inclination angle



Figure 4. Armadillo model, with parameters were $l_{max} = 25$, $\mu = 0.01$, $\mu_1 = 1.0$, $\epsilon = 0.0005$ and $\kappa = 0.0$.

ACKNOWLEDGMENTS

This work was partly supported by the Brazilian Ministry of Science and Technology (MCT/CNPq Universal 02/2006), and the State of Rio de Janeiro (FAPERJ Primeiros projetos 2004).

REFERENCES

- [1] I. Babuska and J. Melenk. The partition of unity method. *International Journal of Numerical Methods in Engineering*, 40:727–758, 1997.
- [2] M. M. Blane, Z. Lei, H. Çivi, and D. Cooper. The 3L algorithm for fitting implicit polynomial curves and surfaces to data. *Transactions on Pattern Analysis and Machine Intelligence*, 22(3):298–313, 2000.
- [3] Y.-L. Chen and S.-H. Lai. A partition-of-unity based algorithm for implicit surface reconstruction using belief propagation. In *Shape Modeling and Applications*, pages 147–155. IEEE, 2007.
- [4] R. Franke and G. Nielson. Smooth interpolation of large sets of scattered data. *International Journal of Numerical Methods in Engineering*, 15:1691–1704, 1980.

- [5] J. P. Gois, V. Polizelli, T. Etienne, E. Tejada, A. Castelo, T. Ertl, and L. G. Nonato. Robust and adaptive surface reconstruction using partition of unity implicits. In *Sibgrapi*, pages 95–104. IEEE, 2007.
- [6] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital Michelangelo project: 3D scanning of large statues. In *Siggraph*, pages 131–144. ACM, 2000.
- [7] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *Pattern Analysis and Machine Intelligence*, 13(11):1115–1138, 1991.
- [8] J. Carr, R. Beatson, J. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *Siggraph*, pages 67–76. ACM, 2001.
- [9] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares. Efficient implementation of Marching Cubes’ cases with topological guarantees. *Journal of Graphics Tools*, 8(2):1–15, 2003.
- [10] M. Alexa, J. Behr, D. Cohen–Or, S. Fleishman, D. Levin, and C. Silva. Point set surfaces. In *Visualization*, pages 21–28. IEEE, 2001.
- [11] M. Alexa, J. Behr, D. Cohen–Or, S. Fleishman, D. Levin, and C. Silva. Computing and rendering point set surfaces. *Transactions on Visualization and Computer Graphics*, 9(1):3–15, 2003.
- [12] T. K. Dey and S. Goswami. Provable surface reconstruction from noisy samples. *Computational Geometry: Theory and Applications*, 35(1–2):124–141, 2006.
- [13] N. Amenta and M. Bern. Surface reconstruction by voronoi filtering. *Discrete and Computational Geometry*, 22(4):481–504, 1999.
- [14] N. Amenta, S. Choi, T. Dey, and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. In *Symposium on Computational Geometry*, pages 213–222. ACM, 2000.
- [15] N. Amenta, S. Choi, and R. Kolluri. The Power Crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications*, 19(2-3):127–153, 2001.
- [16] Y. Ohtake, A. Belyaev, and M. Alexa. Sparse low-degree implicit surfaces with applications to high quality rendering, feature extraction, and smoothing. In *Symposium on Geometry processing*, page 149. Eurographics, 2005.
- [17] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. In *Siggraph*, volume 22, pages 463–470. ACM, 2003.
- [18] T. Tasdizen, J. P. Tarel, and D. B. Cooper. Algebraic curves that work better. In *Computer Vision and Pattern Recognition*, volume II, pages 35–41. IEEE, 1999.
- [19] M. Lage, F. Petronetto, A. Paiva, H. Lopes, T. Lewiner, and G. Tavares. Vector field reconstruction from sparse samples. In *Sibgrapi*, pages 297–304. IEEE, 2006.
- [20] D. Levin. *Mesh-independent surface interpolation*, pages 37–49. Springer, 2003.
- [21] B. Mederos, N. Amenta, L. Velho, and L. de Figueiredo. Surface reconstruction for noisy point clouds. In *Symposium on Geometry Processing*, pages 53–62, 2005.
- [22] V. V. Savchenko, A. Pasko, O. Okunev, and T. Kuni. Function representation of solid reconstructed from scattered surface points and contours. *Computer Graphics Forum*, 14:181–188, 2002.
- [23] A. Sharf, T. Lewiner, G. Shklarski, S. Toledo, and D. Cohen-Or. Interactive topology-aware surface reconstruction. In *Siggraph*, pages 43.1–43.9, 2007.
- [24] I. Tobor, P. Reuter, and C. Schlick. Reconstructing multi-scale variational partition of unity implicit surfaces with attributes. *Graphical Models*, 68(1):25–41, 2006.
- [25] G. Turk and J. O’Brien. Modeling with implicit surfaces that interpolates. *Transaction on Graphics*, 21:855–873, 2002.